Lorenzo Wynberg

Ingeniero de software sénior · Ingeniero de producto con mente de sistemas · orquestación de IA

San José, Costa Rica · Remoto · Permiso de trabajo: UE (ciudadano neerlandés), Costa Rica · Inglés (nativo), español (fluido)

dev.wynberg@gmail.com · +506 8313-8209 · wynberg.dev · github.com/LorenzoWynberg

PERFIL

Desarrollo software profesionalmente desde 2012. Pienso en sistemas: cuando me topo con un problema, construyo lo que resuelve toda esa clase de problemas, y después lanzo con eso. En los últimos 18 meses eso fueron cuatro productos que diseñé y construí de punta a punta, solo: Trinity, un sistema donde un equipo y sus agentes de IA planifican, construyen y lanzan releases completos; una suite de plugins de audio profesional con DSP escrito desde cero en C++; una plataforma de logística con despacho automático; y una app de contabilidad offline-first con cifrado de extremo a extremo para Latinoamérica, España y Portugal. Trabajo investigando: cuando algo es incierto, escribo la prueba, lo mido y diseño a partir de lo que encuentro. Dirijo agentes de IA como un líder dirige un equipo, y construyo los pipelines, las compuertas y las reglas escritas que hacen el trabajo repetible y confiable. Estoy cerca de hacia dónde va el campo y me muevo rápido (todo Trinity migró a Effect 4 tres días después de su versión estable), y construyo mis propias herramientas, como un marketplace público de plugins para Claude Code y Codex.

EXPERIENCIA

Fundador e ingeniero único · Independiente — Mar 2025 – Presente

Costa Rica (remoto) · Cuatro productos diseñados, construidos y lanzados en solitario

Trinity — Entorno de desarrollo agéntico (ADE) — Ene 2026 – Presente

Beta pública · v0.7 en producción · trinityailabs.com

Un solo lugar donde un equipo construye, mantiene y mejora sus productos con IA. Todo gira en torno a releases: los planifica como releases → épicas → historias, corre varios a la vez (features, hotfixes, líneas de mantenimiento, release candidates) en git worktrees aislados, solo hace merge con el CI en verde, y etiqueta y publica cada release por ti.

- Diseñé una jerarquía de hilos de agentes especializados: un Arquitecto que planifica cada release y sigue los releases de los que depende, un agente de release que sigue el plan, corre seis revisiones de release como subagentes con veredictos explicados y clasifica las historias que fallan, y un capataz por historia que corre analista → implementador → auditor → documentador. Personas del equipo (diseñadores incluidos) y agentes trabajan en los mismos hilos, los agentes se avisan entre sí cuando tocan los mismos archivos, y todo lo que espera a una persona cae en una sola lista de pendientes.
- Construí autonomía por niveles, de Supervisado a Desatendido: cada agente corre aislado en un sandbox, y las historias desatendidas tienen su propio agente que concede o niega solicitudes de acceso según la especificación y el plan de la historia; los secretos y las solicitudes de acceso total siempre escalan a una persona, cada concesión queda registrada y es revocable, y los releases se detienen en límites de gasto y de tiempo.
- Construí el motor de ejecución en paralelo y la revisión: git worktrees por historia entre repos, merges solo con CI en verde a pesar de conflictos y fallas parciales, 14 tipos de revisores entre Claude y Codex, y un juez con su propio jurado, que se llama con @judge, que investiga y compara opciones cara a cara.
- Diseñé la arquitectura de una app de escritorio en Tauri v2 (SolidJS, Effect 4), un plano de control en Next.js 16 y sincronización en tiempo real sobre Cloudflare Durable Objects; gestión de releases para los proyectos que entrega, con líneas de mantenimiento, release candidates y publicación en npm/crates.io detrás de una vista previa de aprobación.
- Más de 30 versiones desde abril de 2026, ~800k líneas de TypeScript y ~4,300 archivos de pruebas, construido en solitario. Su última ronda de proyectos de prueba, con configuraciones complejas y escenarios de release construidos y entregados por completo a través de Trinity, produjo más de 450 commits, más de 230 PRs y 74 tags de versión en siete repos, incluidos monorepos con release candidates por paquete.

Ozner Audio — Plugins de audio profesional · C++ / JUCE — Sep 2026 – Presente

En producción · v1.0 · ozneraudio.com

Un sintetizador mono analógico semimodular (Sweet Berta), un rack que encadena hasta 64 unidades de efectos con su propio modo de interfaz de audio independiente (OA Effects), y cada unidad de efecto como su propio plugin de pedal. AU, VST3, LV2 y standalone en macOS, Windows y Linux, más la tienda, el licenciamiento y el instalador alrededor.

- Escribí el DSP desde cero en una librería solo de headers y sin frameworks: oversampling de 2/4/8× con la latencia reportada al host, antialiasing por antiderivadas de primer y segundo orden (incluida una etapa con compensación de retardo dentro del lazo de realimentación de un filtro ladder no lineal), osciladores PolyBLEP y una reverb de resorte con oversampling.
- Diseñé un solo shell de plugin compartido por 39 unidades de efectos y tres productos, con paneles en React dentro de WebViews nativos y un triple buffer sin locks entre los hilos de interfaz y de audio.
- Convertí el gusto en una herramienta: fijé los objetivos del banco de presets a oído y con investigación, y luego construí la medición que los codifica (un binario de niveles offline que lee picos de muestra y true peak y sonoridad BS.1770, cada macro llevada al máximo sola y en conjunto, repartido entre máquinas), para que los agentes puedan diseñar presets nuevos que encajen y lo demuestren. El aliasing, pluginval y pruebas de resistencia son compuertas de cada release.

- Entregué el pipeline comercial: pkg notarizado para macOS, paquetes para Windows y Linux, firma de licencias, una tienda y un gestor de licencias. Un CI consciente del costo manda cada trabajo primero a mis propias máquinas Mac, Windows y Linux, hace cola en ellas cuando están ocupadas, y solo paga runners en la nube cuando todas están apagadas. ~170k líneas de C++.

Mandados — Plataforma de mandados y entregas — Mar 2025 – Presente

Antes del lanzamiento

Mandados, compras y mensajería en Costa Rica para personas, hoteles y empresas, con choferes asalariados.

- Construí el producto completo: API en Laravel 12 / PostGIS, administración en Next.js, apps de cliente y de chofer en React Native (iOS, Android y web), actualizaciones en tiempo real con Reverb y notificaciones con FCM; ~8,900 pruebas.
- Diseñé el despacho automático como un problema de empaquetado: revisiones de factibilidad, ranking de choferes por anillos de distancia, simulación de inserción dentro de ventanas de tiempo y re-despacho en cascada cuando una ventana se desborda.
- Construí el ciclo de pagos detrás de un contrato con el proveedor: tarjetas guardadas, conciliación entre lo cotizado y la ruta real en recargos o reembolsos automáticos, y cobro de deudas.

Kuentas Online — Contabilidad con cifrado de extremo a extremo — Oct 2026 – Presente

En producción · v1.11 · kuentas.online

Contabilidad pensada para el teléfono, para pequeños negocios en Latinoamérica, España y Portugal, en español, portugués e inglés (ventas, inventario, deudas, préstamos); el servidor solo ve texto cifrado.

- Diseñé el protocolo criptográfico sobre libsodium/WebCrypto (X25519, Ed25519, XChaCha20-Poly1305, Argon2id): un log de commits encadenado por hashes que los dispositivos se niegan a bifurcar, una lista de roles firmada a partir de un hash génesis fijado, y compartimentos por rol que el servidor puede retener; quitar a un miembro obliga a rotar las llaves.
- Cualquier moneda local, con tasas en vivo de bancos centrales y fuentes locales (BCE, la PTAX de Brasil, las tasas paralelas de Argentina y Venezuela); offline-first sobre SQLite cifrado en WASM. Lancé la 1.0 dos semanas después del primer commit; ~3,000 pruebas.

Líder de equipo y desarrollador principal · Broccoli Studios · Costa Rica — 2020 – 2026

SABER — Ministerio de Educación Pública (MEP) — 2023 – 2026

Lideré el equipo de Broccoli en la plataforma educativa nacional de Costa Rica, para más de 1 millón de estudiantes y docentes (React/TS, C#/.NET).

- Líder de equipo y desarrollador principal de un equipo multidisciplinario de 5 a 6 personas (desarrolladores, diseñadores, redactores técnicos): planificación, revisión de código, mentoría y releases a producción.
- Punto único de contacto con el ministerio, de principio a fin: requerimientos con la dirección y los analistas del MEP, especificaciones y contratos de API, escenarios de prueba y pruebas de aceptación, entregables y aprobaciones. Responsable de toda la documentación y autor de buena parte de ella (especificaciones, manuales de usuario, notas de versión).
- Entregué más de 8 módulos principales, entre ellos un motor de formularios con SurveyJS, con versionado e instantáneas, que reemplazó flujos en papel, y permisos jerárquicos a nivel nacional, regional e institucional.

Proyectos para clientes

- Lideré proyectos para clientes en React, Laravel y WordPress, desde la arquitectura hasta el despliegue; manejé hosting, DNS, caché y CI/CD. Publiqué laravel-to-i18next, un paquete de Laravel de código abierto.

Pausa profesional — Cuidado familiar — 2016 – 2020

- Cuidador principal de uno de mis padres, junto con pequeños proyectos web independientes. Seguí aprendiendo todo el tiempo: aprendí JavaScript moderno y React, y estudié a fondo diseño de software y patrones de arquitectura.

Desarrollador full-stack · NurseRecruiter · Estados Unidos (remoto) — 2012 – 2016

- Propuse reemplazar el framework PHP propio de la empresa por Laravel, y luego reescribí toda la plataforma junto al desarrollador principal, como un equipo de dos: cuentas, publicación de empleos, paneles para reclutadores, gestión de currículums, búsqueda y mensajería.
- Reconstruí el sistema de emparejamiento en el centro de la plataforma: algoritmos que emparejan enfermeras con hospitales y reclutadores por ubicación, tipo de trabajo, especialidad y más, con esquemas y consultas rediseñados para mantenerlo rápido. Trabajé directamente con los fundadores en arquitectura y producto.

HABILIDADES

Lenguajes y frameworks: TypeScript, C++, PHP, C#, Rust, SQL · React, Next.js, SolidJS, React Native, Effect, Laravel, JUCE

Datos e infraestructura: PostgreSQL/PostGIS, SQLite/libSQL/Turso, Drizzle, WebSockets · Vercel, Cloudflare Workers/Durable Objects/R2, GitHub Actions, Docker, firma de código y notarización

Especialidades: Orquestación de múltiples agentes · criptografía aplicada y sincronización E2E · DSP en tiempo real y plugins de audio · diseño de protocolos sobre Bitcoin (white paper de BAP)

Escritos: 22 artículos de ingeniería sobre orquestación de agentes, cifrado, pruebas, rendimiento y arquitectura (en inglés) · trinityailabs.com/blog